(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
5 December 2002 (05.12.2002)

PCT

(10) International Publication Number
WO 02/098100 A1

(51) International Patent Classification[7]: H04L 29/06, G06F 1/00, H04L 29/12

(21) International Application Number: PCT/GB01/02417

(22) International Filing Date: 31 May 2001 (31.05.2001)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant *(for all designated States except US)*: PREVENTON TECHNOLOGIES LIMITED [GB/GB]; Hanover House, Hanover International Conference Centre, Reading, Berkshire RG30 3UN (GB).

(72) Inventor; and

(75) Inventor/Applicant *(for US only)*: ROUX, Peter, Terrance [ZA/GB]; 41 Winnall Manor Road, WInchester SO23 0NW (GB).

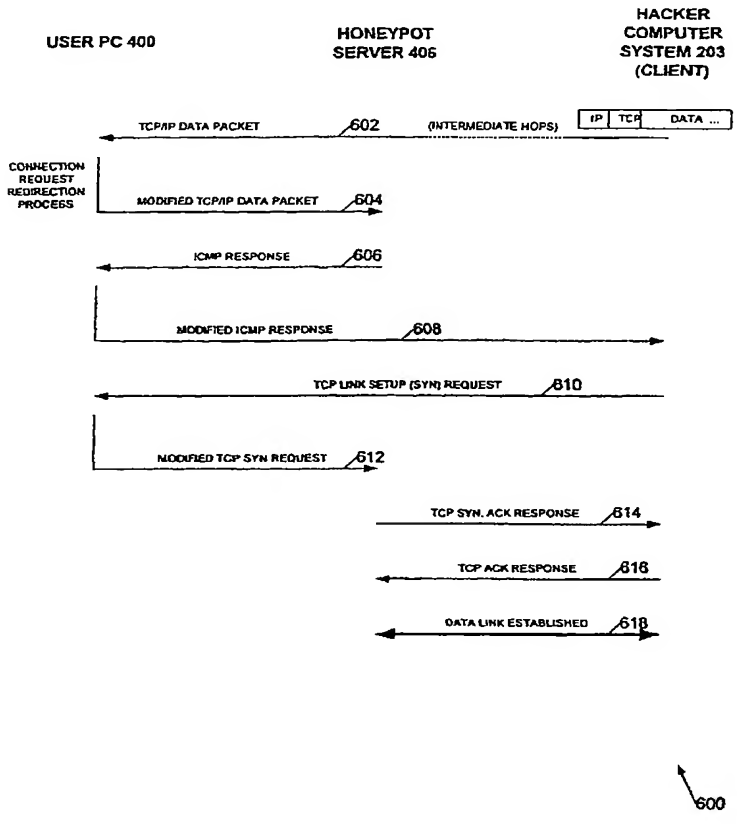
(74) Agent: COLLINS, John, David; Marks & Clerk, 57-60 Lincoln's Inn Fields, London WC2A 3LS (GB).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) Title: ACCESS CONTROL SYSTEMS



(57) Abstract: A system, method, and computer software for controlling access to computer systems by third parties such as hackers. Software (402) detects attempeted accesses to ports of open server processes on a computer (400) and redirects such attempted accesses, preferably transparently. The hacker may be redirected to a honeypot server (406) arranged to resemble the target computer, to deceive the hacker into thinking the attempted access was successful. The software improves computer security and can assist in monitoring and trapping hackers.

WO 02/098100 A1

Published:
— *with international search report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

ACCESS CONTROL SYSTEMS

Field of the Invention

The present invention generally relates to the control of access to computer systems by remote third parties. More particularly it relates to computer software, methods, and systems for restricting access to open ports of processes running on a computer to protect against, for example, attacks by hackers.

Background to the Invention

The increasing sophistication of computers, particularly computers for home and small business use, increases the likelihood of leaving vulnerable backdoors of which hackers may take advantage. Sometimes such computers (or "PCs") store credit card data or other confidential information, but even where a user is not particularly concerned about the confidentiality of data stored on a computer, the computer may nonetheless be employed by a hacker as the basis for an attack on another computer system. It is therefore generally desirable to restrict the access of hackers to such computers.

A computer is particularly vulnerable to attack by a hacker when it is connected to a computer network such as the Internet. Communication over the Internet uses internet protocol (IP), under which data is transmitted in data packets to which IP adds an IP header. The IP header contains an IP source address, an IP destination address, checksum data, and other data such as time to live (TTL) data. The time to live data comprises a number which is decremented by one each time the IP packet passes through a router, the data packet being discarded when the number reaches zero. This prevents endless loops and also allows route tracing by incrementally increasing the time to live value of packets and monitoring how far they get.

Every computer connected to the Internet has an internet or IP address, which is presently a 32 bit address arranged as four eight bit octets, for example 129.1.48.254. Internet Service Providers (ISPs) are generally allocated a range of internet addresses which they can allocate to users. An IP address may be allocated by an ISP to a subscribing user either temporarily, for example on dial-up, or permanently. Where a user accesses the Internet using an ADSL (asymmetrical digital subscriber line) modem or other "always on" technology a permanent IP address is normally allocated. Internet resources such as web pages are located by means of uniform resource locators (URLs) which specify, among other things, a hostname of a computer on which the resource is stored. The URL is translated into an IP address by means of a domain name server or server tree.

A range of additional protocols are used over the internet protocol, the most common of which is Transmission Control Protocol (TCP). Other protocols running over IP include UDP (User Datagram Protocol), and RDP (Reliable Data Protocol) - a more complete list can be found in RFC 1700. The transmission control protocol breaks up data for communication over the Internet into a number of packets or datagrams, reassembles the datagrams in the correct order and performs error checking after transmission. The internet protocol is responsible for routing the individual datagrams. Like IP, TCP adds a header to each datagram including, among other things, source and destination port number data, sequence number data (for reassembling the datagrams) and a selection of flags. The flags include a Synchronize Sequence Numbers (SYN) flag which synchronizes sequence numbers to begin a connection, an Acknowledgement Field Significant (ACK) flag which, when set, indicates that the datagram includes acknowledgement data for confirming reception, and a Finish (FIN) flag which, which set, indicates that all the data has been sent. Depending upon the amount of data to be sent, one or more datagrams may be necessary.

An IP packet 100 including a TCP datagram is shown schematically in Figure 1. The IP data packet comprises an IP header 102, and a TCP header 104, as described above, and data 106.

TCP/IP is used to provide services available over the Internet such as a File Transfer Protocol (FTP), a Network Terminal Protocol (TELNET), a Simple Mail Transfer Protocol (SMTP), and other protocols such as the HyperText Transfer Protocol (HTTP) which is used for providing Hypertext Markup Language (HTML) documents on the World Wide Web. These services are provided as client/server services, a server providing services for other computers coupled to the network.

When a connection is made to a computer specified by an IP address, a service may be requested by specifying a port. A number of port numbers are conventionally assigned to specific services – for example FTP uses ports 20 and 21, and HTTP uses port 80. On home computers port 139 provides NetBIOS session services for performing network operations on the computer.

A connection is established between client and server processes using pairs of sockets, each socket having a socket address comprising an internet address and a TCP port number. Under TCP a server process socket listens for a connection request and a client requests a connection to the socket. Once two sockets have been connected data can be transmitted bidirectionally and transparently. This is known as stream communication; other protocols, such as UDP, use datagram communication in which the socket address of the receiving socket is sent in each datagram. Generally although one socket will have an assigned port number, the other socket of the connection will have a port number selected more or less at random from those port numbers which are not reserved (ports 1024 and upwards). Programs to determine what ports on a computer are open for access are widely available for network administration purposes.

Figure 2 shows, schematically, connection of a user terminal, PC 200, to a server 202 via an internet service provider 204 over the Internet.

User PC 200 comprises operating system software 206 and application layer software 208. The operating system software includes point-to-point protocol (PPP) software 201 and a TCP IP stack 212. The application layer software includes a web browser 214 such as Microsoft Internet Explorer (Registered Trade Mark), and other application programs 216 such as MS Word, MS Excel, games software such as Doom,

communications software such as PC Anywhere (Trade Mark), e-mail software, NetBIOS software, and the like. The application layer software communicates with the operating system software for communication and other purposes (for ease of illustration other details of the operating system software are not shown).

PC 200 is connected to a modem 218, which may be internal, which in turn is coupled to a telephone line 220 for connecting PC 200 to the Internet. Modem 218 may be a conventional audio modem or an ADSL modem or, where PC 200 is connected to the Internet via a cable TV link, a cable modem. Modem 218 communicates with a similar modem 222 coupled to Internet service provider computer system 204; in general modem 222 will be one of many ISP modems.

ISP computer system 204 includes PPP software 224, which communicates with PPP software 210 in the user PC to allow a modem connection to be established between the user PC 200 and the ISP computer system 204. The ISP computer system also includes a TCP/IP stack 226 and, once the modem connection has been established, TCP/IP stacks 212 and 226 establish a TCP/IP communications link through which further communications between PC 200 and ISP 204 take place.

ISP computer system 204 is also coupled to internet communications network 201 by means of one or more high bandwidth fixed lines 228. Also coupled to internet communications network 201 is server computer system 202, also including a TCP/IP stack 230, as well as web server code 232 and web page data 234. The ISP computer system 204 includes DNS software 236 for resolving URLs, such as a URL for server 202, to IP addresses.

Figure 2 also illustrates a hacker computer system 203, connected to internet 201 for hacking into PC 200. The hacker computer system again includes a TCP/IP stack 238, as well as application software 240 such as port scan software and other software usable for hacking.

Figure 3 shows, in outline, data communications involved in setting up a TCP/IP link between user PC 200 and ISP 204 and retrieving web page data from server 202. Figure 3 also shows outline details of a hacker attack on PC 200.

An initial exchange 300 sets up a TCP/IP connection between PC 200 and ISP 204. This comprises an PPP communication 302 from PC 200 to establish a TCP/IP link 304. In the illustrated example the TCP/IP connection links to web browser 214 and thus uses port 80 on PC terminal 200, at the user's IP address, and an arbitrarily allocated port number at the ISP IP address. For this connection web browser 214 is the client.

Once a TCP/IP link has been established between PC terminal 200 and ISP 204 web browser 214 can retrieve web page data from server 202, as illustrated by data communications 306: browser 214 sends an HTTP request 308 including the URL of server 202 to ISP computer system 204. At the ISP DNS code 236 resolves the URL to an IP address (normally by accessing one or more external servers) and then forwards HTTP request 310 to the server. Web server process 232 on server 202 responds to the HTTP request by sending HTTP data 312, comprising HTML web page data, to the IP address of PC 200 via ISP 204.

Generally speaking, at any one time the average home user has four or five open ports providing server processes which a hacker may access. For example Microsoft Internet Explorer (Trade Mark) processes hyperlinks with a .doc or .xls extension to provide an "Open or Save As" dialogue box which will open Word or Excel as a server for the remote document. Similarly games or communications software running on a PC will often provide a server socket process. Microsoft Windows (Registered Trade Mark) also runs a number of server processes by default, such as NetBIOS (which provides a server socket port 139), and the Windows 2000 printer server process. These may be exploited by a hacker, for example by means of command line buffer overflow.

Data communications 314 illustrate, in outline, connection of hacker computer system 203 to user PC 200. Initially a request 316 to connect to an open socket of a server process on PC 200 is sent to an open port at the PC IP address. The hacker may either scan PC 200 for open ports or simply try those ports which are commonly left open.

Following this a TCP/IP link 318 with the hacker computer system 203 is established. Once this communications link has been established commands and code may be uploaded and data on PC 200 may be examined and/or modified. User PC 200 may also be used by the hacker for a further attack on another computer system. For example a distributed denial of service attack may be made on an internet service provider, a large number of separate home PCs being arranged to simultaneously access the ISP to bring its computer systems to a halt.

Establishing a TCP/IP connection to a user PC 200 can take some time and may therefore be difficult where a PC is only allocated an IP address on connection to an ISP. However, with the growth of "always on" technology, and the tendency to assign permanent IP addresses to users, the vulnerability to hacking is increased.

It is known to provide personal firewall software to check for incoming connection requests on specified ports and then block such requests, but this approach has a number of problems. Generally such software is complicated and difficult to set up, involving the definition of a number of rules which can interact. Furthermore such firewalls prevent a hacker from connecting to a socket by simply blocking an open port, which can alert a hacker to the presence of sensitive information. In combination with the difficulty of closing down all the potential holes in a system such personal firewalls can, in some instances, create more problems that they solve.

There therefore exists a need to provide improved software and methods for restricting access of hackers to computers, and particularly to home and small business computers. There exists a further need to provide improved software and methods for monitoring such attempted attacks.

Summary of the Invention

According to the present invention, there is therefore provided a carrier medium carrying computer readable code for controlling a computer to restrict access to said computer, said computer being couplable to a network and, when coupled to the network, having a network address; the code comprising: code to receive an incoming

data packet including destination address data specifying the network address of said computer; code to modify said destination address data to specify a modified network address; and code to send said data packet to said modified network address; whereby said incoming data packet is redirected from the network address of said computer to said modified network address.

The carrier medium may be a storage medium such as a hard or floppy disk or CD-ROM, or an optical or electrical signal carrier.

By redirecting the incoming data packet on from its supposed final destination a hacker may be deceived into thinking that the message correctly arrived at its destination without being alerted to the presence of access control software. The data packet may either be diverted to a non-responding address, in which case the hacker may assume that a port connection is not in fact present, or the data packet may be directed to another computer such as a "honeypot" server. The purpose of such a "honeypot" server is to deceive a hacker into thinking that a connection with the target computer has been made when in fact the connection has been diverted to another computer, preferably configured to resemble, at least superficially, the main features a hacker would expect to find on the target machine. Such a "honeypot" server may also be used to monitor a hacker's activities and, in some instances, to trap a hacker.

In a preferred embodiment the incoming data packet comprises internet protocol data including a destination internet address which is modified to redirect the data packet. If no response is desired the data packet may be redirected to an impermitted or invalid IP address, such as a broadcast address (for example 255.255.255.255) or a reserved address. Alternatively the data packet may be redirected to the IP address of another computer such as a honeypot server. The honeypot server can be implemented on the target computer but for security it is preferably implemented on a different machine.

An IP data packet from a hacker or other undesirable source will generally include data for communicating with or connecting to a server process of the computer. The code may be configured to handle a TCP, UDP, RDP, or other protocol data packet. In the case of TCP, UDP and RDP protocols the data packet further comprises port number

data for specifying a port number of the computer server process. Thus the code preferably further comprises code to modify the port number data to redirect the data packet to a port at the modified (IP) network address. This allows a server process on, for example, a honeypot server to handle the incoming data packet and provide a spoof response to a hacker. It may not be necessary to modify the port number of the incoming data packet where, for example, a server process with the same port number is set up on a honeypot server to handle the data packet. However it is desirable to provide for modification of the port number, for greater flexibility.

In many environments the code can make use of an existing TCP/IP stack with hooks into dynamically linked library routines. In a preferred embodiment a mapping table is included for mapping an incoming request to connect to a data port to a modified (IP) address and, optionally, a modified port number. Thus such a table may comprise a plurality of entries each comprising a port number for an incoming data connection request, and a corresponding redirection IP address and, optionally, redirection port number. Preferably a user interface is also provided to allow the data in this table to be set up. This provides a simple and readily understandable way of setting up redirection rules. It will be appreciated that with this arrangement there is no need to redirect all incoming requests to connect to a (server) port and, if desired, a user may select some ports for mapping to a redirected address, and others to remain open.

In a preferred embodiment the redirection process leaves the time to live data or "hop count" of the incoming (IP) data packet unchanged. This provides an additional layer of security against an attempt to trace the route the packet has taken and hence detect the redirection.

In one embodiment an additional option is provided to automatically close a connection to a port after the link has been inactive for a predetermined time and/or a predetermined time after the cessation of reception of incoming data. Normally the TCP FIN flag is set to indicate the end of a data stream, and by setting a time-out a predetermined interval after the non-appearance of a FIN flag, denial of service attacks can be helped to be prevented.

In another aspect the invention provides a carrier medium carrying computer readable code for redirecting a transmission control protocol/internet protocol (TCP/IP) request for connection to a server process, the code comprising: code for receiving at an IP destination address an incoming TCP/IP connection request to connect to a server process at the IP destination address, the TCP/IP connection request having IP header data including the IP destination address; code for modifying at least said IP destination address of said TCP/IP connection request; and code for forwarding the said TCP/IP connection request to said modified address.

In a further aspect the invention provides a method of restricting access to a computer coupled to a computer network, the computer having a computer network address, the method comprising: receiving at the computer, from the network, a data packet having said computer network address as a destination address; modifying said destination address of said data packet; and passing said modified data packet to a data transmission process for forwarding said data packet to said modified address.

In a still further aspect the invention provides a method of redirecting a transmission control protocol/internet protocol (TCP/IP) request for connection to a server process, the method comprising: receiving an incoming TCP/IP connection request including IP header data having an IP destination address; modifying at least said IP destination address of said TCP/IP connection request; and forwarding the said TCP/IP connection request to said modified address.

In another aspect the invention provides a method of restricting access to a communications socket of a computer program, the communications socket having an address comprising a port number and a network address, the method comprising: receiving, at the network address, data intended for the computer program communications socket; modifying the received data to specify an alternative socket address; and redirecting the data to the alternative socket address.

Preferably the alternative socket address comprises the socket address of a computer program running on another machine such as a honeypot server, configured to resemble

the target computer but with false, misleading, or spoof information. This machine may be used to monitor access to the communication socket, for example to trap the hacker.

In a further aspect the invention provides a method of managing attempted accesses to a target computer system, the method comprising: receiving, at the computer system, a request for an internet protocol (IP) session with the computer system from a remote user; and redirecting the request to a second computer system.

Preferably the redirection is substantially transparent to the remote user so that without careful monitoring the redirection is not apparent to the remote user.

The above-described methods may be implemented by computer program code stored, for example, on a computer readable medium such as a disk. The code may comprise source or executable code, and may be written in any conventional computer language.

In a yet further aspect the invention provides computer processing apparatus comprising: a data memory operable to store data to be processed; an instruction memory operable to store one or more application programs providing a network server function, and storing processor implementable code; a processor operable to process the data in accordance with the stored code; a network interface for coupling the apparatus to a network; and wherein the stored processor implementable code further comprises: code to receive an incoming data packet including destination address data specifying the network address of said computer; code to modify said destination address data to specify a modified network address; and code to send said data packet to said modified network address; whereby said incoming data packet is redirected from the network address of said computer to said modified network address to restrict access to a said application program server function.

Brief Description of the Drawings

These and other aspects of the invention will now be further described, by way of example only, with reference to the accompanying figures in which:

Figure 1 shows a TCP/IP data packet;

Figure 2 shows a schematic block diagram of a user terminal connected to a server via an internet service provider;

Figure 3 shows an outline data communications diagram for the arrangement of Figure 2, illustrating a hacker attack;

Figure 4 shows a schematic diagram of a computer network including a computer with access control software, and a honeypot server;

Figure 5 shows an exemplary user interface for an access control program;

Figure 6 shows a data flow diagram for redirection of an incoming TCP/IP connection request from a hacker computer; and

Figure 7 shows a flowchart for the redirection process of Figure 6.

Detailed Description of Preferred Embodiments

Referring to figure 4, the server 202, hacker computer 203, ISP 204 and Internet 201 correspond to those illustrated in figure 2. User terminal (PC) 400 comprises a general purpose computer system, similar to PC 200 of figure 2, onto which additional access control software 402 has been loaded. Thus computer 400 stores operating system program code 210, 212 for implementing PPP and TCP/IP protocols, and application code 214, 216 for a web browser and other application programs. Computer 400 is connectable to a modem 218 for connection to Internet 201 via ISP 204 over a telephone line or other data link. Computer 400 also includes volatile and non-volatile data storage, a keyboard, mouse, and display and other conventional components not shown in figure 3.

Access control software 402 is additionally loaded onto computer 400. This software may be provided to computer 400 on a removable storage device 404, such as a floppy

disk, or this software may be downloaded from a server over the internet. The access control software 402 comprises an access control user interface 402a coupled to a network device driver 402b. User interface 402a allows the network device driver software 402b to be configured to redirect incoming data packets, such as TCP/IP data packets from hacker computer system 402 requesting set-up of a socket connection to application program 216 or to operating system programs running on computer 400.

The access control software 402 operates to re-direct incoming TCP/IP requests, as described in more detail below. When installed, the network device driver 402b in effect provides a filter function for incoming IP data. In a Windows (registered trade mark) PC system, network device driver 402b is set up as the default device driver for network communications (where multiple device drivers are present in a Windows system, each is tried in turn by the system). Device driver 402b runs as part of the kernel of the computer system, and is accessed by applications via kernel drivers which are part of the operating system software.

The network device driver may be implemented using a Microsoft DDK (device driver kit) such as the 98 DDK (for Windows 95 and 98) or the NT DDK (for Windows NT and 2000). Implementing the device driver is relatively straightforward as use may be made of existing operating system routines such as DLLs (Dynamically Linked Library routines) to provide the necessary low level functions such as packet send, packet receive, packet get address, packet open adapter, get adapter name, and the like. In other operating systems similar use may be made of dynamic linking, for example using shared object files under SunOS/Solaris (registered trade marks) operating systems.

In one embodiment implemented in a Windows environment, the network device driver is primarily implemented at layer 3 of the Windows operating system, although a flag is set to indicate incoming data at Windows layer 0. In a preferred embodiment, an error message is displayed if another device driver is installed in promiscuous mode as this could, under some circumstances, allow another application to take precedence over access to a port.

Referring again to Figure 4, a honeypot server computer system 406 is, optionally, also provided. Like computer 400, computer system 406 comprises a general purpose computer suitably programmed. Thus honeypot server 406 is couplable to a modem 408 for connecting the server to Internet 201 and, like computer 400, includes PPP operating system code 410 and TCP/IP operating system code 412. At least one server process 414 is also loaded onto the honeypot computer system to allow hacker system 203 to connect to honeypot server 406.

Preferably, the honeypot server is loaded with a mirror of the applications on user PC 400, and thus includes, for example, a web browser such as browser 214 and other application programs corresponding to the set of application programs loaded onto user PC 400. Spoof data (not shown in Figure 3) may also be loaded onto honeypot server 406 to simulate what a hacker might expect to see when connecting to user PC 400. However, although the same broad categories of data may be loaded onto the honeypot server confidential data is preferably not placed on this server. Broadly speaking, the idea is to re-direct a request from hacker computer system 203 to connect to user PC 400 to the honeypot server 406, without making the hacker aware of the redirection. The hacker will therefore imagine he or she is connected to user PC 400, and data is loaded onto honeypot server 406 to maintain this subterfuge. In fact the hacker will have been directed away from sensitive data to a computer system on or from which relatively little harm can be done.

Optionally, monitoring software 416 may be loaded onto honeypot server 406 to monitor a hacker's actions. This monitoring software should preferably be difficult to find or substantially invisible to the hacker; the monitoring software may attempt to determine details of the hacker computer system 203. Where a hacker is to be trapped by the honeypot server, server 406 may be provided with an easy to break password and data of potential interest to a hacker, such as spoof password files, spoof credit card details and the like.

Figure 5 shows an exemplary user interface 500 for access control software 402.

In Figure 5 a user is presented with a table comprising three columns. A Port In column 502, an IP (address) Out column 504, and a Port Out column 506. The Port In column is used for defining open ports to server processes on computer 400 which are to be redirected, and the IP (address) Out and Port Out columns are for entering an IP address and port to which data packets for the ports to be protected are to be redirected.

As shown in Figure 5, an incoming TCP IP request to connect to a server socket on user PC 400 at port 80 (corresponding to web browser 214) is redirected to IP address 1.2.3.4, although the port number is unchanged as a hacker would expect to make a connection to port 80 as this is a well-known, assigned port number. Similarly a request to user PC 400 to connect to NetBIOS using a server socket at port 139 is also directed to port 139 at IP address 1.2.3.4. Preferably IP address 1.2.3.4 is the IP address of honeypot server 406, onto which are loaded web browser and NetBIOS application programs to give the impression that a connection to user PC 400 has been established when, in fact, the connection is to honeypot server 406.

Referring again to Figure 5, an incoming FTP request (port 21) is redirected to IP address 255.255.255.0 which is a broadcast address and will not therefore allow a socket connection to be established. In this case the port number (port 1) is arbitrary. In the final row of the illustrative redirection mapping table of Figure 5 port 1025 is redirected to port 1035 at the honeypot server's IP address of 1.2.3.4. Port 1025 of user PC 400 may correspond to one of the other application programs 216 installed on the PC, such as a game or Microsoft Word, and this is redirected to the appropriate port for the corresponding application on honeypot server 406. The port number is not conserved since it is not one of the pre-assigned "well-known" port numbers.

To redirect a TCP/IP data packet only one of the IP destination address and port number need be changed and it is therefore theoretically possible to redirect an incoming data packet having the IP address of user PC 400 as its destination IP address to a different port on the same computer (user PC 400). This is generally undesirable as the hacker is given access to user PC 400, albeit in a controlled manner. Nevertheless there may be situations where this is desirable, providing careful control over a hacker's permitted activity is exercised. For example, in some circumstances it may not be practicable to

provide a honeypot server on a separate machine, although it may nonetheless be desirable to monitor a hacker's attempted activities. A drawback with this approach is the potential visibility of the port to which incoming data packets are being redirected to a port scanning program, since the port to which packets are redirected must itself be open.

As can be seen from the above description, the redirection mapping table provides a relatively straightforward user interface for defining redirection rules for redirecting incoming TCP/IP data packets. The user interface is implemented by access control user interface software component 402a, which creates a redirection or mapping table storing corresponding data, for use by network device driver 402b in redirecting incoming data packets.

In the arrangement of Figure 4 the user PC 400 and honeypot server 406 are shown as having separate connections to ISP 204. However in alternative embodiments PC 400 and server 406 may both be connected to a local area network (LAN) or wide area network (WAN) which in turn is connected to an internet service provider via a gateway also coupled to the network.

Referring now to Figure 6, this shows signalling in a TCP/IP handshaking process 600 used to establish a connection between a client hacker computer system 203 and a honeypot server 406 when an incoming connection request from hacker computer system 203 is initially directed to user PC 400.

Initially, at 602, a first TCP/IP data packet is sent from hacker computer system 203 to the hacker's target, user PC 400. This initial TCP/IP data packet has the general form indicated in Figure 1, and has an IP header specifying, among other things, an IP source address (the IP address of hacker computer system 203), an IP destination address (the IP address of user PC 400), time to live data, and a protocol number specifying that the IP data packet includes TCP data. Also in the TCP/IP data packet is a TCP header including, among other things, a source port number (the source port of hacker computer system 203), a destination port number (specifying the port number of an open server process running on user PC 400, although in some circumstances this port

number will not yet have been defined), sequence number data, acknowledgement number data, and a number of flags. In some circumstances a hacker may attempt to gain illicit access to a computer system by providing TCP/IP data packets which are not in accordance with agreed standards, and preferably the access control software 402 is capable of handling such data packets.

On receipt of the initial TCP/IP data packet 602, network device driver 402b modifies at least the IP header data of the incoming data packet to change the destination address to the IP address of honeypot server 406. The modified TCP/IP data packet 604 is then sent to honeypot server 406. This initial TCP/IP data packet contains connection initialization information in the TCP header for establishing a connection between the hacker computer system and another computer system.

Once the honeypot server 406 has received the modified data packet 604, it provides an ICMP (internet control message protocol) response 606 within an IP data packet having the IP address of user PC 400 as its destination address. The network device driver 402b on user PC 400 then modifies the destination address in the IP header to substitute the IP address of hacker computer system 203, and a modified ICMP response 608 is then sent to the hacker computer system.

ICMP returns a "0" to indicate success delivery of a TCP/IP data packet, and returns a "3" to indicate an undeliverable data packet. The ICMP protocol may also return a redirect instruction to indicate that future packets should be sent to an alternative IP address. This may be used internally by access control software 402 but, as will be appreciated, a "redirect" ICMP response is preferably not returned to hacker computer system 203 as such a redirection response would be readily discoverable.

Once the modified ICMP response has been received by hacker computer system 203, this system issues a TCP link set-up request 601, in accordance with the usual TCP three-way handshaking procedure. This link set-up request has the TCP SYN flag set and will usually include a datagram sequence number (for ordering received datagrams). This TCP SYN request is again packaged in an IP data packet having the IP address of user PC 400 as the IP header destination address. This IP packet is received by user PC

400 and network device driver 402b again modifies the destination address to that of honeypot server 406 and forwards a TCP SYN request 612 to honeypot server 406.

In the aforementioned redirection processes implemented by network device driver 402b on user PC 400, where the redirection mapping table (as illustrated in Figure 5) includes a modified port number, the network device driver also modifies the destination port number data in the TCP datagram header (there are no port numbers in ICMP messages).

Following reception of TCP SYN request 612 by honeypot server 406, the honeypot server issues a conventional TCP response 614 comprising an ACK flag, to acknowledge the SYN flag from hacker computer system 203, and its own SYN flag to establish a connection with hacker computer system 203. Generally, associated with the ACK flag, is acknowledge number data indicating the next sequence number the server is expecting. Finally hacker client computer system 203 issues its TCP ACK response 616 to the TCP SYN request 614 from honeypot server 406, completing the three-way TCP handshake and establishing a data link between hacker computer system 203 and honeypot server 406. The TCP SYN, ACK response 614 and TCP ACK response 616 are both packaged within IP data packets, but at this stage the IP header for data packets exchanged between honeypot server 406 and hacker system 203 uses the IP address of honeypot server 406 rather than the IP address of user PC 400. Thus when data link 618 is established, packets of IP data exchanged over this link have source and destination IP addresses of the honeypot server 406 and of the hacker computer system 203, rather than being redirected via user PC 400.

By the time TCP responses 614 and 616 are being exchanged between the hacker computer system 203 and the honeypot server 406 the TCP/IP data link has already been at least partially established and the fact that the TCP responses are now enclosed within IP data packets which are going directly to and from the honeypot server 406 is not readily apparent to most software, although it could be detected by examining the data at a low level such as the IP level. Thus in alternative embodiments the TCP/IP data packets may always be routed via user PC 400, by modifying the source or destination address of the IP packet headers as necessary, optionally also modifying the

TCP header port number data and, preferably, maintaining the IP packet header time-to-live data unchanged.

Normally when a TCP/IP data packet is redirected the time-to-live (TTL) value indicating the number of hops that the packet is allowed to take before it is discarded, is decremented by 1. However in the above-described redirection process the TTL value is preferably unmodified to conceal the redirection from the hacker.

In some circumstances large TCP datagrams may be fragmented at the IP level, and this is indicated by a fragmentation flag in the IP header. In some embodiments of the network device driver 402b this fragmentation flag may be inspected to determine whether or not an IP data packet carries data which has already been checked. If a determination has been made as to whether or not a (fragmented) packet is to be redirected, the (fragmented) packet need not be checked again.

A loophole in the TCP specification can cause a TCP handshaking processing to hang if a TCP/IP data packet is sent specifying a non-existent source IP address. The Windows implementation of TCP times-out after 75 seconds but this time-out can be circumvented by issuing more than 6 non-standard TCP/IP data packets. These holes can be used to mount a denial of service attack on a system. The TCP Finish control flag FIN is usually set to request normal termination of a TCP connection in the direction the datagram containing the flag is travelling (one FIN in each direction is required to completely close a connection). An optional additional feature of access control software 402 therefore monitors the status of a TCP connection and where a FIN flag is not provided following a predetermined time-out period of inactivity, the connection can be closed. This assists in countering such denial of service attacks. The time-out period may be initiated by cessation of data reception or by inactivity of a TCP connection.

Referring now to Figure 7, this shows a flow diagram for the redirection process of figure 6 implemented by the network device driver 402b of access control software 402.

The process illustrated in figure 7 is initialized, at step S10, by reception of an incoming TCP/IP data packet requesting connection to a port of a server process running on user PC 400. At step S11 the access control software reads the destination port number of the TCP header and compares this against incoming port number entries in a redirection table such as the redirection mapping table illustrated in figure 4. If, at step S12, there is no entry for the destination port number in the table the procedure continues to step S13, and the TCP/IP data packet is permitted to be processed by the server process identified by the TCP destination port number, and the procedure then ends at step S14. Processing of the data packets by the identified server process is, in effect, a default state of the access control software which, by not modifying the incoming data packet, allows the data to be processed in the normal way.

If an entry for the destination port number is found in the table the procedure continues to step S15 and the IP address of hacker computer system 203 is stored for later use. Then, at step S16, the new destination IP address and, optionally, the new port number for the data packet is retrieved from the redirection table. The new destination IP address could be an address such as a broadcast address from which no reply would be provided or an invalid or non-existent IP address such as an (as yet) unused class D or E internet address (addresses from 224 onwards) but for the purposes of illustration the new destination IP address will be assumed to be the IP address of honeypot server 406, and the new port number that of a server process running on the honeypot server to give the hacker the impression that access to user PC 400 has successfully been achieved.

Thus, at step S17, the access control code replaces the destination IP address and, optionally, port number of the incoming TCP/IP data packet with the new destination IP address and port number of the honeypot server and server process. Then, at step S18, the modified TCP/IP data is forwarded to the honeypot server and, at step S19, an ICMP response is received back from the honeypot server by user PC 400.

The ICMP response is contained within an IP data packet and, at step S20, the source IP address in the IP header of the data packet is replaced with the IP address of the user PC 400 and, at step S21, the destination IP address in the IP header is replaced with the IP address of hacker computer system 203 (stored in step S15). The, at step S22, the

modified ICMP data packet is forwarded to hacker computer system 203, appearing to come from user PC 400 rather than from honeypot server 406. In response the hacker computer system 203 issues a TCP SYN request which, at step S23, is received by user PC 400. At step S24 the access control software on user PC 400 replaces the destination IP address and, optionally, port number in the received IP packet containing the TCP SYN request, with the IP address and port number of a server process running on honeypot server 406. The modified TCP SYN request (in an IP data packet) is then forwarded, at step S25, to the honeypot server for processing by the honeypot server TCP/IP stack 412.

The honeypot server then continues with the conventional TCP handshaking procedure as described with reference to figure 6 above to complete the establishment of data link 618 between the hacker computer system 203 and the honeypot server 406. The access control software on user PC 400 however awaits receipt of a further incoming TCP/IP data packet requesting connection to a user PC server process (step S26) and on detection of such a packet, at step S27, loops back to starting step S10. In other embodiments, following step S25 the subsequent TCP signals, TCP SYN, ACK response 614, and TCP ACK response 616, and the TCP/IP data link 618 are all passed through user PC 400 by substitution of source and destination IP addresses and port numbers as appropriate to provide a still greater degree of security.

Although the above redirection processes have been described with reference to the TCP transmission control protocol, they are also applicable to other IP-based protocols such as RDP and UDP, and to non-IP-based data transmission where such data transmission operates using packetized data including packet address data.

In the particular case of UDP (user datagram protocol) a very similar process to that described above may be employed for redirecting UDP-containing IP data packets. The UDP protocol uses an IP address and port number for connecting to a server process socket in a similar way to the TCP protocol, although unlike TCP each UDP datagram includes a socket descriptor so that, in effect, each datagram is processed separately. Similarly considerations also apply to RDP (reliable data protocol). Thus the above-described access control software may readily be configured to handle either TCP,

UDP, or RDP data, redirecting incoming data based upon destination port number as described above. Again, as described above, the redirection may be to a non-existent IP address or to an IP address and port number of another server process on another machine such as honeypot server 406.

The access control software has been described in the context of a user personal computer connected to the Internet via a modem connection to an internet service provider but the software may also be employed on a computer connected to the Internet via a LAN or WAN and gateway. The access control methods described above are not restricted to Windows-based environments, but may also be used with other operating systems such as Linux, Unix, the Mackintosh Operating System, OS/2, and other operating systems. Likewise the above-described software and methods are not limited to use on IBM PCs, but may be used on other general-purpose micro or mini computers, on workstations and more powerful machines, and on servers in general. The invention may also be applied to internet-enabled devices incorporating computers connected or connectable to the Internet or another IP network, such as internet-enabled domestic appliances, internet-enabled vending machines, and the like.

The invention is not limited to use with the Internet, and it may be used with other IP networks such as an Intranet or Extranet, as well as with non-IP networks. For example, the invention may be employed with 3G web-enabled mobile phones and with PDAs (personal digital assistants). The underlying idea of redirection of a data packet may, for example, be employed with Bluetooth (Registered Trade Mark) enabled devices which have network addresses.

No doubt many effective alternatives will occur to the skilled person and it will be understood that the invention is not limited to the described embodiments and encompasses modifications apparent to those skilled in the art lying within the spirit and scope of the claims appended hereto.

CLAIMS:

1. A carrier medium carrying computer readable code for controlling a computer to restrict access to said computer, said computer being couplable to a network and, when coupled to the network, having a network address; the code comprising:

code to receive an incoming data packet including destination address data specifying the network address of said computer;

code to modify said destination address data to specify a modified network address; and

code to send said data packet to said modified network address;

whereby said incoming data packet is redirected from the network address of said computer to said modified network address.

2. A carrier medium carrying computer readable code as claimed in claim 1, wherein said incoming data packet comprises an internet protocol (IP) data packet and wherein said computer network address and said modified network address comprise internet addresses.

3. A carrier medium carrying computer readable code as claimed in claim 2, wherein said IP data packet further includes port number data for specifying a port number of a server process of said computer, and wherein said code further comprises:

code to modify said port number data for redirecting said incoming data packet to a modified port number at said modified network address.

4. A carrier medium carrying computer readable code as claimed in claim 2, wherein said IP data packet further includes port number data for specifying a port number of said computer; wherein said computer readable code includes mapping table data for mapping a said port number of said computer to data specifying a said modified network address; and wherein said code further comprises:

code to read said port number data of said IP data packet; and

code to use said port number data to retrieve said modified network address from said mapping table data.

5. A carrier medium carrying computer readable code as claimed in claim 4, wherein said mapping table data further includes data for mapping a said port number of said computer to a modified port number at said modified network address; and wherein said code further comprises:

code to use said port number data to retrieve said modified port number from said mapping table data; and

code to modify said port number data for redirecting said incoming data packet to a modified port number at said modified network address.

6. A carrier medium carrying computer readable code as claimed in claim 2, wherein said IP header data includes time to live (TTL) data related to the number of systems through which the IP packet has passed, and wherein said time to live data of the data packet sent to the modified network address and said time to live data of said incoming data packet define the same number of systems.

7. A carrier medium carrying computer readable code as claimed in claim 2, wherein said IP data packet includes transmission control protocol (TCP) header data for establishing a connection with a server process of said computer.

8. A carrier medium carrying computer readable code as claimed in claim 7, wherein said TCP header data includes a FIN flag for indicating completion of data transmission and wherein said code includes code to disconnect a connection with a said server process after a predetermined period of connection inactivity when a FIN flag has not been received.

9. A carrier medium carrying computer readable code as claimed in claim 1, wherein said modified network address comprises a network address from which no valid reply will be provided to the redirected data packet.

10. A carrier medium carrying computer readable code as claimed in claim 1, wherein said code further comprises:

code to provide a user interface for defining a said modified network address.

11. A carrier medium carrying computer readable code for redirecting a transmission control protocol/internet protocol (TCP/IP) request for connection to a server process, the code comprising:

code for receiving at an IP destination address an incoming TCP/IP connection request to connect to a server process at the IP destination address, the TCP/IP connection request having IP header data including the IP destination address;

code for modifying at least said IP destination address of said TCP/IP connection request; and

code for forwarding the said TCP/IP connection request to said modified address.

12. A carrier medium carrying computer readable code as claimed in claim 11, wherein said incoming TCP/IP connection request includes port data specifying a TCP/IP connection request port number and wherein said code further comprises code for modifying said port data to specify a modified port number, for forwarding said TCP/IP connection request to a server process having said modified port number on a computer at said modified address.

13. A carrier medium carrying computer readable code as claimed in claim 12, wherein said computer readable code further comprises:
user interface code for associating a port number of a said TCP/IP connection request with a said modified port number and a said modified address; whereby a user may set up one or more TCP/IP connection request redirection rules.

14. A carrier medium carrying computer readable code as claimed in claim 11, wherein said incoming TCP/IP connection request includes TCP time to live (TTL) data and wherein the TTL data of said incoming TCP/IP connection request is the same as the TTL data of said forwarded TCP/IP connection request.

15. A method of restricting access to a computer coupled to a computer network, the computer having a computer network address, the method comprising:

receiving at the computer, from the network, a data packet having said computer network address as a destination address;

modifying said destination address of said data packet; and

passing said modified data packet to a data transmission process for forwarding said data packet to said modified address.

16. A method of restricting access to a computer as claimed in claim 15, wherein said data packet comprises a request for connection to a server process on the computer.

17. A method of restricting access to a computer as claimed in claim 16, wherein said modified address comprises a network address from which no valid response is provided in response to reception of the data packet.

18. A method of restricting access to a computer as claimed in claim 16, wherein said modified address comprises a network address of a computer storing program code and/or data to resemble the computer to which access is restricted.

19. A method of restricting access to a computer as claimed in claim 15, wherein said data packet comprises an internet protocol (IP) data packet having header data including a hop count, and wherein the hop count of said modified data packet is equal to or greater than the hop count of said received data packet.

20. A method of redirecting a transmission control protocol/internet protocol (TCP/IP) request for connection to a server process, the method comprising:

receiving an incoming TCP/IP connection request including IP header data having an IP destination address;

modifying at least said IP destination address of said TCP/IP connection request; and

forwarding the said TCP/IP connection request to said modified address.

21. A method of restricting access to a communications socket of a computer program, the communications socket having an address comprising a port number and a network address, the method comprising:

receiving, at the network address, data intended for the computer program communications socket;

modifying the received data to specify an alternative socket address; and

redirecting the data to the alternative socket address.

22. A method as claimed in claim 21, wherein the alternative socket address comprises the socket address of a second computer program; and

further comprising storing the said computer program on a first computer, and storing said second computer program on a second computer configured to resemble said first computer insofar as the configuration of said second computer is practicably determinable using the socket address of the second computer.

23. A method as claimed in claim 22, further comprising using the second computer to monitor access to the communications socket.

24. A method of managing attempted accesses to a target computer system, the method comprising:

receiving, at the computer system, a request for an internet protocol (IP) session with the computer system from a remote user; and

redirecting the request to a second computer system.

25. A method as claimed in claim 24, wherein said redirection is substantially transparent to said remote user.

26. A computer readable medium carrying computer readable data to, when running, implement the method of any one of claims 15, 20, 21, 24 and 25.

27. Computer processing apparatus comprising:

a data memory operable to store data to be processed;

an instruction memory operable to store one or more application programs providing a network server function, and storing processor implementable code;

a processor operable to process the data in accordance with the stored code;

a network interface for coupling the apparatus to a network; and

wherein the stored processor implementable code further comprises:

code to receive an incoming data packet including destination address data specifying the network address of said computer;

code to modify said destination address data to specify a modified network address; and

code to send said data packet to said modified network address;

whereby said incoming data packet is redirected from the network address of said computer to said modified network address to restrict access to a said application program server function.

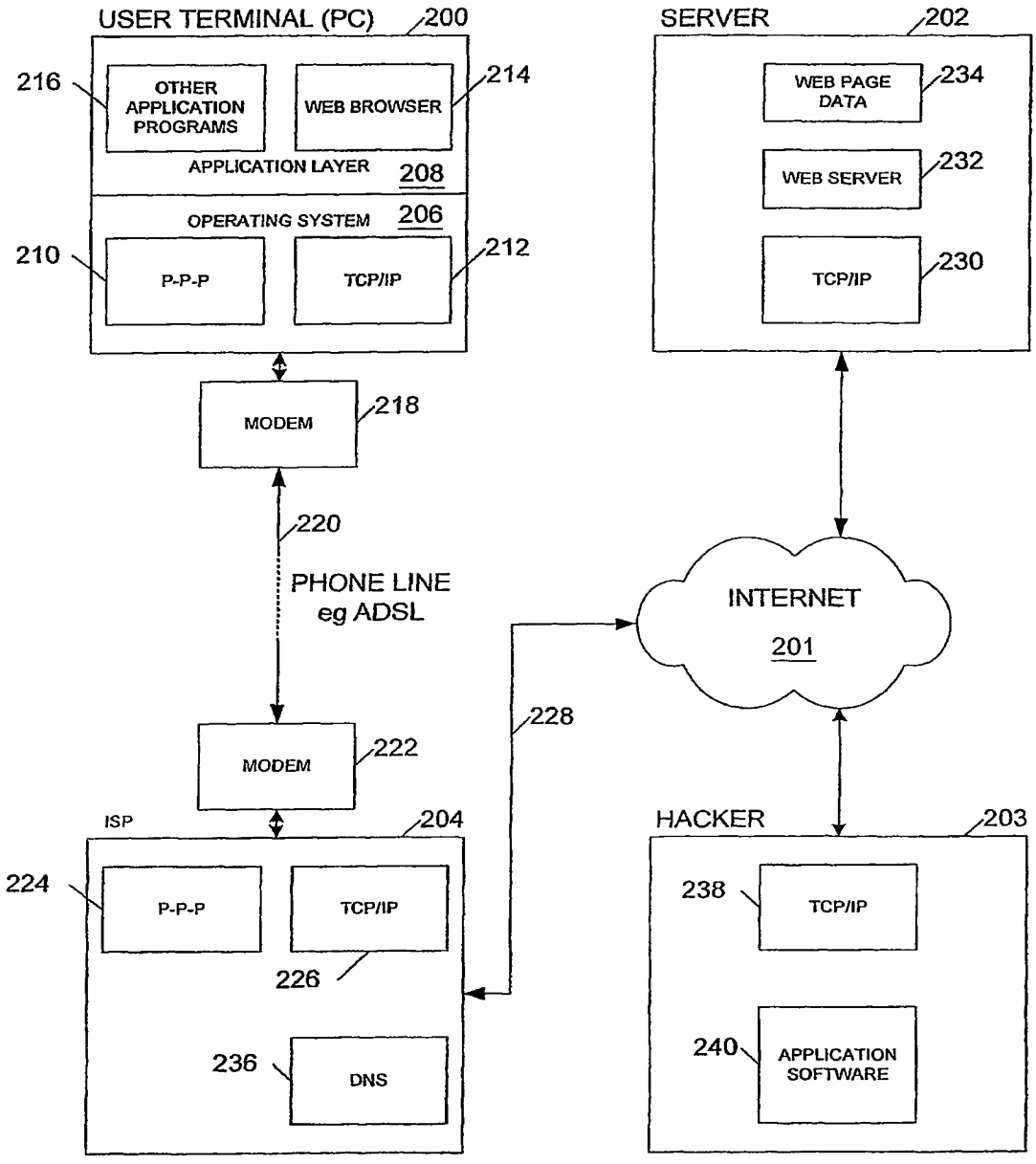
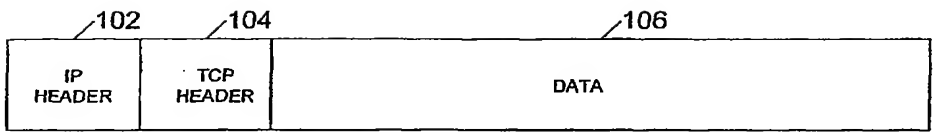


Figure 2
(PRIOR ART)



100

Figure 1
(PRIOR ART)

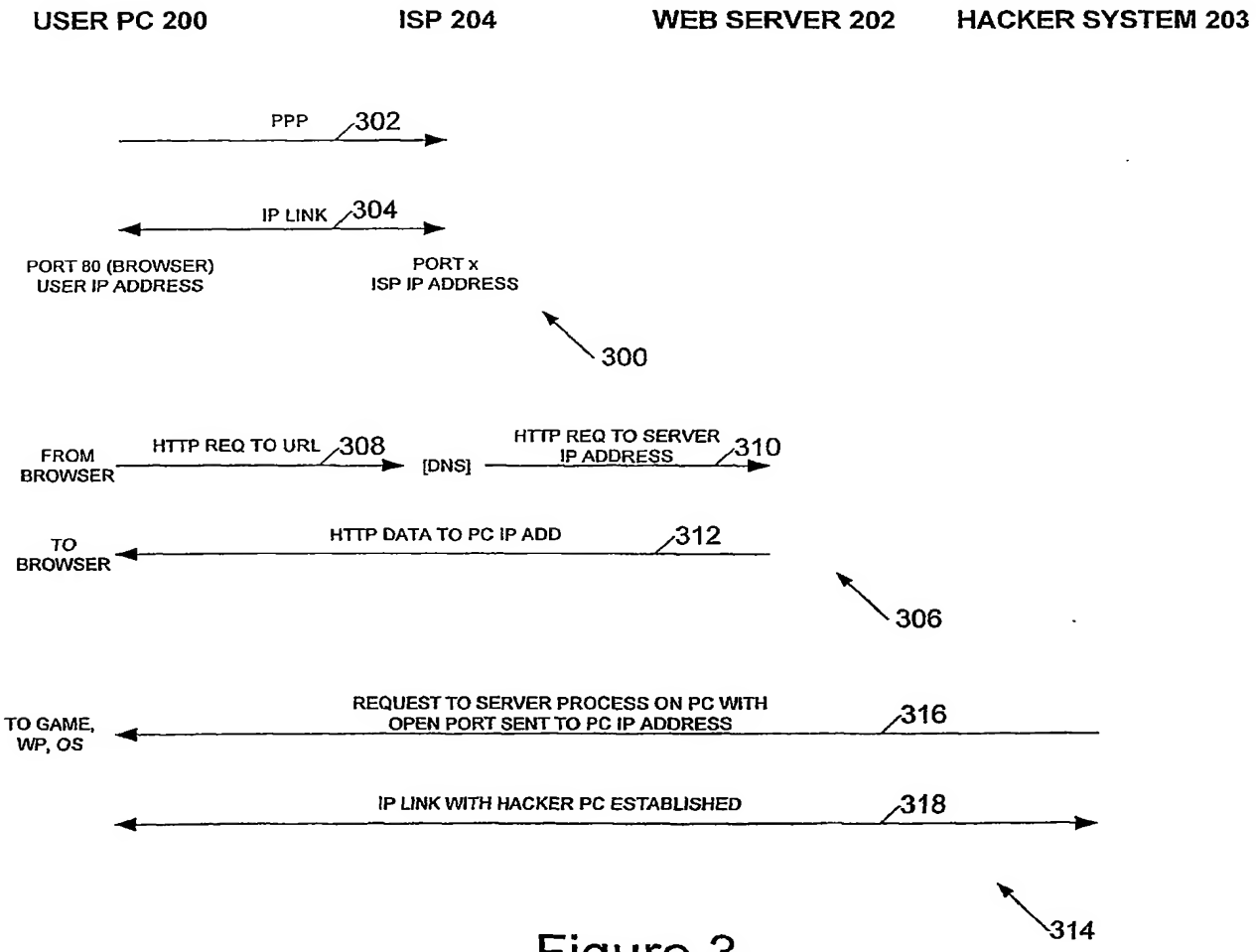


Figure 3
(PRIOR ART)

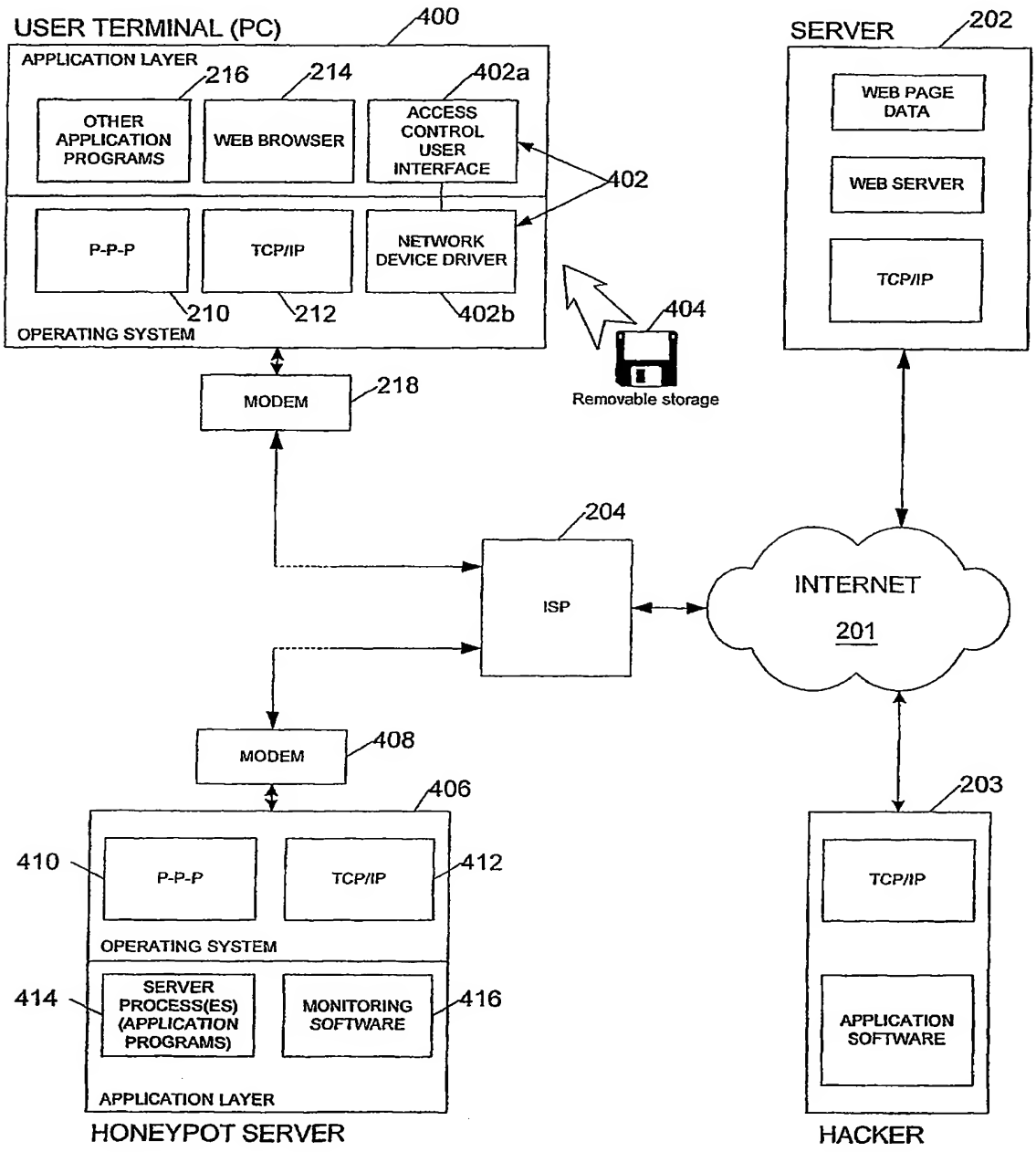
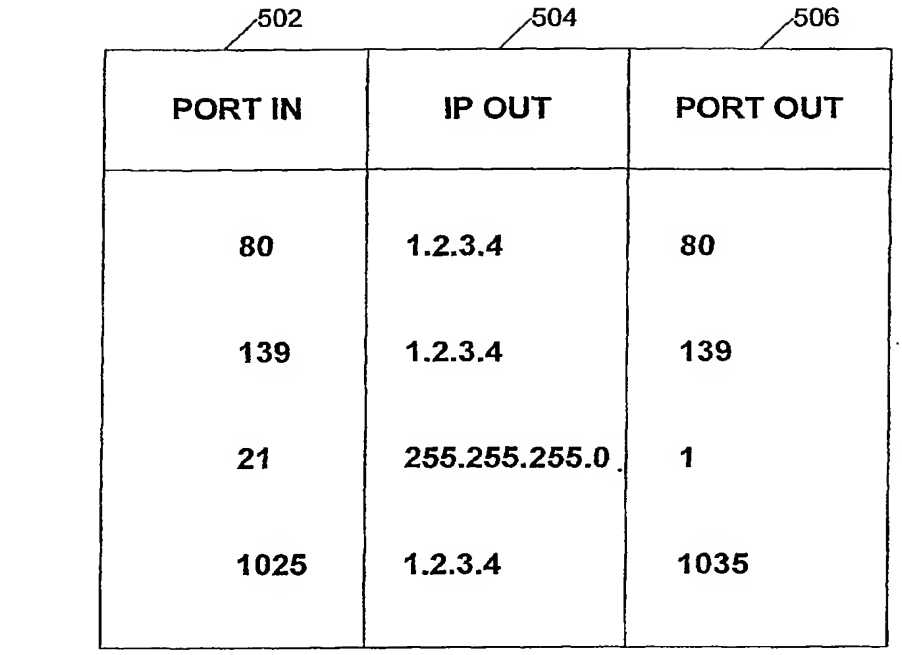

USER TERMINAL (PC)
400
APPLICATION LAYER
216
214
402a
OTHER APPLICATION PROGRAMS
WEB BROWSER
ACCESS CONTROL USER INTERFACE
402
P-P-P
TCP/IP
NETWORK DEVICE DRIVER
210
212
402b
OPERATING SYSTEM
404
Removable storage
MODEM
218
SERVER
202
WEB PAGE DATA
WEB SERVER
TCP/IP
204
ISP
INTERNET
201
MODEM
408
406
203
410
P-P-P
TCP/IP
412
OPERATING SYSTEM
414
SERVER PROCESS(ES) (APPLICATION PROGRAMS)
MONITORING SOFTWARE
416
APPLICATION LAYER
TCP/IP
APPLICATION SOFTWARE
HONEYPOT SERVER
HACKER

Figure 4



PORT IN	IP OUT	PORT OUT
80	1.2.3.4	80
139	1.2.3.4	139
21	255.255.255.0	1
1025	1.2.3.4	1035

Figure 5

500

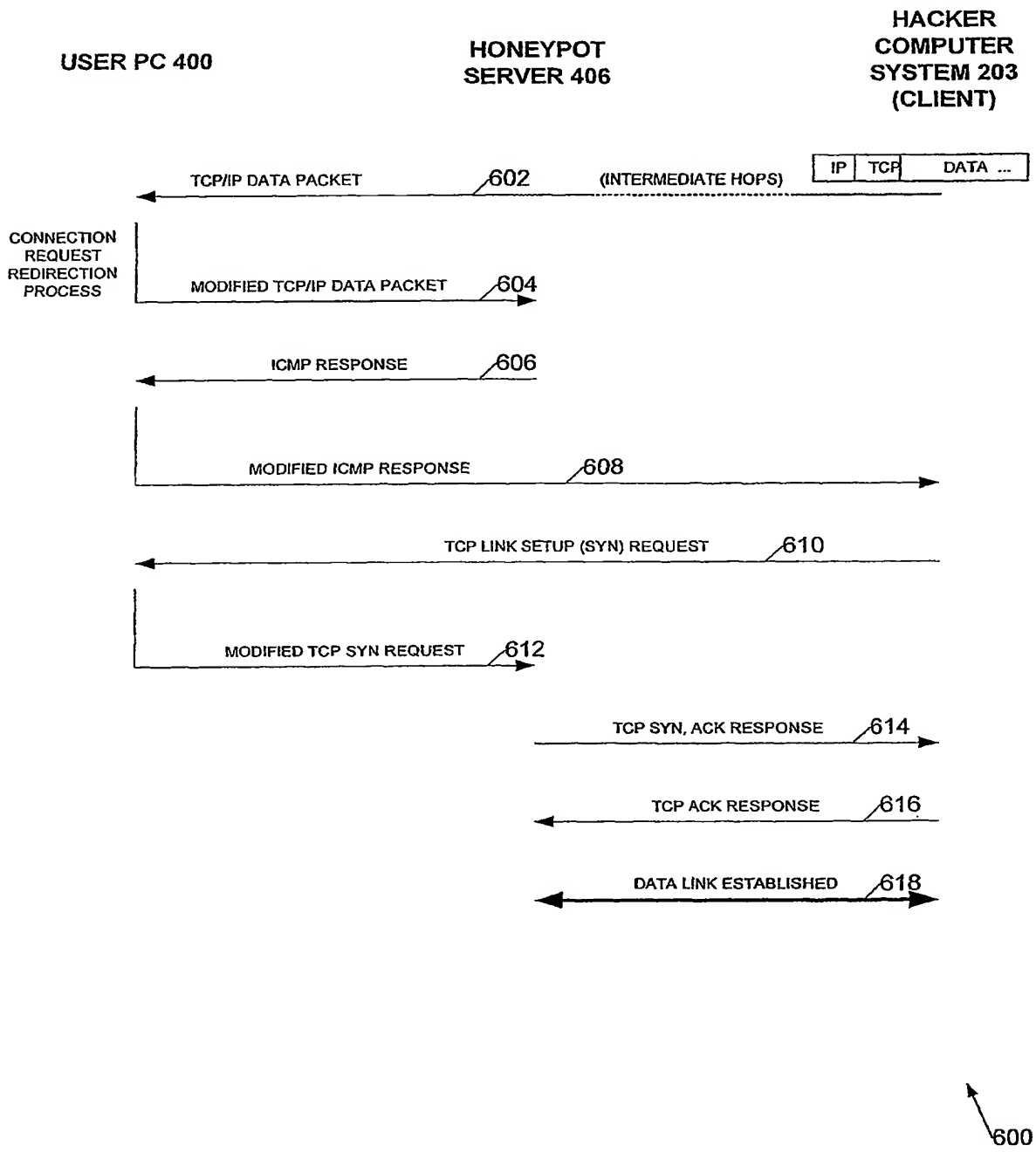

USER PC 400
HONEYPOT SERVER 406
HACKER COMPUTER SYSTEM 203 (CLIENT)
IP
TCP
DATA ...
TCP/IP DATA PACKET
602
(INTERMEDIATE HOPS)
CONNECTION REQUEST REDIRECTION PROCESS
MODIFIED TCP/IP DATA PACKET
604
ICMP RESPONSE
606
MODIFIED ICMP RESPONSE
608
TCP LINK SETUP (SYN) REQUEST
610
MODIFIED TCP SYN REQUEST
612
TCP SYN, ACK RESPONSE
614
TCP ACK RESPONSE
616
DATA LINK ESTABLISHED
618
600

Figure 6

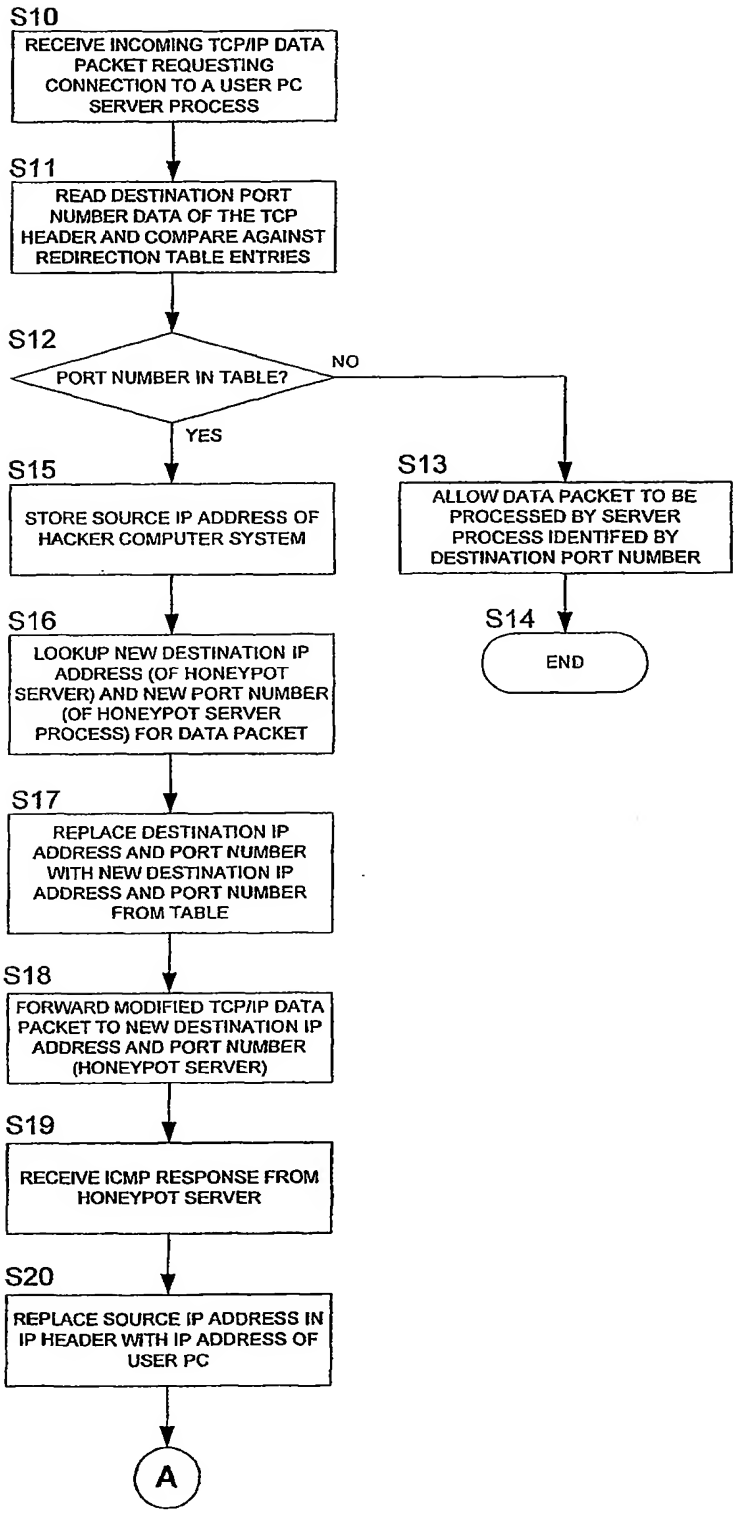

S10
RECEIVE INCOMING TCP/IP DATA PACKET REQUESTING CONNECTION TO A USER PC SERVER PROCESS
S11
READ DESTINATION PORT NUMBER DATA OF THE TCP HEADER AND COMPARE AGAINST REDIRECTION TABLE ENTRIES
S12
PORT NUMBER IN TABLE?
NO
YES
S13
ALLOW DATA PACKET TO BE PROCESSED BY SERVER PROCESS IDENTIFED BY DESTINATION PORT NUMBER
S14
END
S15
STORE SOURCE IP ADDRESS OF HACKER COMPUTER SYSTEM
S16
LOOKUP NEW DESTINATION IP ADDRESS (OF HONEYPOT SERVER) AND NEW PORT NUMBER (OF HONEYPOT SERVER PROCESS) FOR DATA PACKET
S17
REPLACE DESTINATION IP ADDRESS AND PORT NUMBER WITH NEW DESTINATION IP ADDRESS AND PORT NUMBER FROM TABLE
S18
FORWARD MODIFIED TCP/IP DATA PACKET TO NEW DESTINATION IP ADDRESS AND PORT NUMBER (HONEYPOT SERVER)
S19
RECEIVE ICMP RESPONSE FROM HONEYPOT SERVER
S20
REPLACE SOURCE IP ADDRESS IN IP HEADER WITH IP ADDRESS OF USER PC
A

Figure 7a

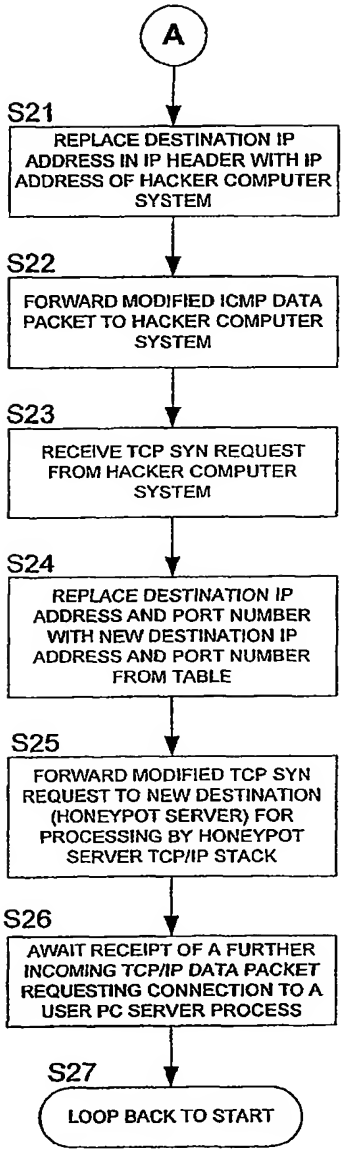

A
S21
REPLACE DESTINATION IP ADDRESS IN IP HEADER WITH IP ADDRESS OF HACKER COMPUTER SYSTEM
S22
FORWARD MODIFIED ICMP DATA PACKET TO HACKER COMPUTER SYSTEM
S23
RECEIVE TCP SYN REQUEST FROM HACKER COMPUTER SYSTEM
S24
REPLACE DESTINATION IP ADDRESS AND PORT NUMBER WITH NEW DESTINATION IP ADDRESS AND PORT NUMBER FROM TABLE
S25
FORWARD MODIFIED TCP SYN REQUEST TO NEW DESTINATION (HONEYPOT SERVER) FOR PROCESSING BY HONEYPOT SERVER TCP/IP STACK
S26
AWAIT RECEIPT OF A FURTHER INCOMING TCP/IP DATA PACKET REQUESTING CONNECTION TO A USER PC SERVER PROCESS
S27
LOOP BACK TO START

Figure 7b

INTERNATIONAL SEARCH REPORT

Interr ıal Application No
PCT/GB 01/02417

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04L29/06 G06F1/00 H04L29/12

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal, WPI Data, PAJ

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 6 148 336 A (BROCK KEVEN J ET AL) 14 November 2000 (2000-11-14)	1-5,7, 9-13, 15-18, 20-27
A	column 9, line 62 -column 10, line 26 column 10, line 52 -column 11, line 13 column 11, line 60 -column 12, line 7 column 12, line 16-22 column 12, line 49-55 figures 11,12,14	6,8,14, 19

-/--

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
18 March 2002	25/03/2002

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Lázaro, M.L.

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Inten nal Application No
PCT/GB 01/02417

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
X	US 6 061 349 A (HOWES RICHARD A ET AL) 9 May 2000 (2000-05-09)	1-4,7,9, 11-13, 15-18, 20,24-27
A	abstract column 2, line 44 -column 3, line 4 column 4, line 46 -column 6, line 14 column 7, line 31-52 column 10, line 50-67 figures 1,2A,2B,4B	5,8,10, 14,19, 21-23
A	EP 1 011 244 A (LUCENT TECHNOLOGIES INC) 21 June 2000 (2000-06-21) abstract page 3, line 16 -page 4, line 5 page 4, line 33 -page 5, line 5 page 5, line 21-32 page 12, line 12-23	1-27

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

ıation on patent family members

Intern nal Application No
PCT/GB 01/02417

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 6148336 A	14-11-2000	US 6141686 A	31-10-2000
US 6061349 A	09-05-2000	US 5793763 A	11-08-1998
		US 6324177 B1	27-11-2001
		US 6108300 A	22-08-2000
		US 5989060 A	23-11-1999
		US 6298063 B1	02-10-2001
		US 6317775 B1	13-11-2001
		US 6104717 A	15-08-2000
EP 1011244 A	21-06-2000	EP 1011244 A2	21-06-2000